

25-25-06

JFW

CERTIFICATE OF MAILING BY "EXPRESS MAIL" (37 CFR 1.10)

Applicant(s): Modha, et al.

Docket No.

AM9990184US2

Application No.

10/660,242

Filing Date

September 11, 2003

Examiner

Apu M. Mofiz

Customer No.

29154

Group Art Unit

2165

Invention:

CLUSTERING HYPERTEXT WITH APPLICATIONS TO WEB SEARCHING

I hereby certify that this Declaration Under 37 C.F.R. 1.131

(Identify type of correspondence)

is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 in an envelope addressed to: Director of the United States Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450 on

May 23, 2003

(Date)

Tylene McCoy

(Typed or Printed Name of Person Mailing Correspondence)

(Signature of Person Mailing Correspondence)

EV 815252268 US

("Express Mail" Mailing Label Number)

Note: Each paper must have its own certificate of mailing.



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re patent application of:
Modha et al.

Serial No.: 10/660,242

Filed: September 11, 2003

Group Art Unit: 2165

Examiner: Mofiz, Apu M.

Atty. Docket No.: AM9990184US2

For: CLUSTERING HYPERTEXT WITH APPLICATIONS TO WEB SEARCHING

Commissioner of Patents
P.O. BOX 1450
Alexandria, VA 22313-1450

DECLARATION UNDER 37 C.F.R. §1.131

We, the inventors of the invention defined by claims 28-40 and 55-61 of U.S. Patent Application Serial No. 10/660,242 hereby declare the following:

[0001] The purpose of this declaration is to prove that we conceived the claimed invention prior to the earliest effective prior art date of Kuo et al., "Web Document Classification based on Hyperlinks and Document Semantics," which is presently understood to be August 2000. The following shows that we conceived our invention prior to August 2000 and that we were diligent from our date of conception to its reduction to practice and were further diligent to the date of the filing of our patent application, which is a divisional patent application based on U.S. Patent No. 6,684,205 having a priority date (filing date) of October 18, 2000 (hereinafter, the parent patent application is referred to as the "Patent Application").

[0002] We are all the inventors of the subject matter claimed in claims 28-40 and

55-61 of U.S. Patent Application Serial No. 10/660,242.

[0003] During all time periods mentioned herein, and specifically between our conception date and the filing date of the Patent Application, all activities described herein occurred in the United States.

[0004] Proof of the conception of the claimed invention prior to August 2000, and diligence in reducing the invention to practice and filing the Patent Application is demonstrated in the attached Exhibits, labeled as Exhibits A and B.

[0005] As shown in Exhibit A, which is an invention disclosure form, we conceived the claimed invention at a date prior to August 2000. As permitted by MPEP 715.07, the dates on Exhibit A have been removed; however, we hereby declare that all dates thereon are prior to August 2000. Further, the invention was actually conceived before Exhibit A was prepared. Therefore, our conception date actually predates Exhibit A.

[0006] Exhibit A generally discloses the claimed invention. For example, independent claims 28, 55, and 61 recite, “searching a database of documents,” which is generally described on the bottom of page 1 of Exhibit A.

[0007] As shown in Exhibit B, which is a technical research report paper, we conceived the claimed invention at a date prior to August 2000. The technical research report paper provided in Exhibit B was presented at the IMA (Institute for Mathematics and its Applications) “Hot Topics” Workshop: Text Mining, April 17-18, 2000, University of Minnesota; Minneapolis, Minnesota and was later presented at the Proceedings of ACM Hypertext Conference, May 30-June 3, 2000, San Antonio, Texas. In fact, the invention was actually conceived before Exhibit B was prepared.

[0008] Exhibit B specifically discloses the claimed invention. As provided in

independent claims 28, 55, and 61, “searching a database of documents,” is disclosed on page 143, column 2, fourth paragraph (paragraph beginning, “Throughout the paper...”) through page 144, column 1, line 2 of Exhibit B. As provided in independent claims 28, 55, and 61, “performing a search of said database using a query to produce query result documents,” is disclosed on page 143, column 1, lines 1-3 of Exhibit B. As provided in independent claims 28, 55, and 61, “constructing a word dictionary of words within said query result documents,” is disclosed on page 144, column 1, last line through page 144, column 2, first line of Exhibit B. As provided in independent claims 28, 55, and 61, “constructing an out-link dictionary of documents within said database that are pointed to by said query result documents; adding said query result documents to said out-link dictionary,” is disclosed on page 144, column 2, second paragraph (paragraph beginning, “Out-links We now outline the creation...”) of Exhibit B. As provided in independent claims 28, 55, and 61, “constructing an in-link dictionary of documents within said database that point to said query result documents; and adding said query result documents to said in-link dictionary,” is disclosed on page 144, column 2, fifth paragraph (paragraph beginning, “In-links The creation of *B*...”) of Exhibit B.

[0009] As provided in dependent claims 29 and 56, “forming first vectors for words remaining in said word dictionary,” is disclosed on page 144, column 2, first paragraph (paragraph beginning, “Suppose d unique words remain...”) of Exhibit B. As provided in dependent claims 29 and 56, “forming second vectors for documents remaining in said out-link dictionary,” is disclosed on page 144, column 2, fourth paragraph (paragraph beginning, “Suppose f unique nodes remain...”) continuing through Figure 1 of Exhibit B. As provided in dependent claims 29 and 56, “forming third vectors for documents remaining in said in-link dictionary,” is disclosed on page 145, column 1, first paragraph (paragraph beginning, “Suppose b unique nodes remain...”) of Exhibit B. As provided in dependent claims 29 and 56, “normalizing said first vectors, said second vectors, and said third vectors to create vector triplets for document remaining in said in-link dictionary and said out-link dictionary,” is disclosed on page 145, column 1, first paragraph (paragraph beginning, “Normalization Finally, for each

document...”) of Exhibit B. As provided in dependent claims 29 and 56, “clustering the said vector triplets into one of clusters, or classes and partitions,” is generally disclosed on page 145, column 2, sixth paragraph (paragraph beginning, “Concept Triplets Suppose we are given n document vector...”) of Exhibit B.

[0010] As provided in dependent claims 30 and 57, “said clustering comprises a four *toric k-means* process comprising arbitrarily segregating the vector triplets into clusters; for each cluster, computing a set of concept triplets describing said cluster; re-segregating said vector triplets into a more coherent set of clusters by putting each vector triplet into a cluster corresponding to a concept triplet that is most similar to, a given vector triplet; determining a coherence for each of said clusters based on a similarity of vector triplets within each of said clusters, and repeating the computing and re-segregating steps until coherence of the obtained clusters no longer significantly increases,” is disclosed on page 145, column 2, fourth paragraph (section beginning, “TORIC *k*-MEANS ALGORITHM...”) through page 147, column 1, fourth paragraph, of Exhibit B.

[0011] As provided in dependent claims 31 and 58, “annotating and summarizing said clusters using nuggets of information, said nuggets including summary, breakthrough, review, keyword, citation, and reference,” is disclosed on page 143, column 2, second paragraph (paragraph beginning, “We annotate each cluster generated by the *toric k-means*...”) of Exhibit B.

[0012] As provided in dependent claims 32 and 59, “wherein said summary comprises a document in a cluster having a most typical in-link feature vector amongst all the documents in said cluster,” is disclosed on page 147, column 1, fifth paragraph (paragraph entitled, “summary”) of Exhibit B.

[0013] As provided in dependent claims 33 and 59, “wherein said breakthrough comprises a document in a cluster having a most typical in-link feature vector amongst all

the documents in said cluster,” is disclosed on page 147, column 1, sixth paragraph (paragraph entitled, “breakthrough”) of Exhibit B.

[0014] As provided in dependent claims 34 and 59, “wherein said review comprises a document in a cluster having a most typical out-link feature vector amongst all the documents in said cluster,” is disclosed on page 147, column 1, seventh paragraph (paragraph entitled, “review”) of Exhibit B.

[0015] As provided in dependent claims 35 and 59, “wherein said keyword comprises a word in said word dictionary for said cluster that has the largest weight,” is disclosed on page 147, column 1, eighth paragraph (paragraph entitled, “keywords”) of Exhibit B.

[0016] As provided in dependent claims 36 and 59, “wherein said citation comprises a document in a cluster representing a most typical in-link into said cluster,” is disclosed on page 147, column 1, ninth paragraph (paragraph entitled, “citations”) of Exhibit B.

[0017] As provided in dependent claims 37 and 59, “wherein said reference comprises a document in a cluster representing a most typical out-link out of said cluster,” is disclosed on page 147, column 2, first paragraph (paragraph entitled, “references”) of Exhibit B.

[0018] As provided in dependent claims 38 and 60, “pruning function words from said word dictionary,” is disclosed on page 144, column 2, lines 1-6 of Exhibit B.

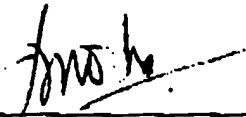
[0019] As provided in dependent claims 39 and 60, “pruning documents from said out-link dictionary that are pointed to by fewer than a first predetermined number of said query result documents,” is disclosed on page 144, column 2, third paragraph, lines 5-7 (paragraph beginning, “To treat nodes in...” of Exhibit B.

[0020] As provided in dependent claims 40 and 60, “pruning documents from said in-link dictionary that point to fewer than a second predetermined number of said query result documents,” is disclosed on page 144, column 2, lines 1-6 of Exhibit B and on page 144, column 2, sixth paragraph, (paragraph beginning, “To treat nodes in...”) through page 145, column 1, lines 1-5, of Exhibit B.

[0021] We were diligent from the date of conception in reducing the invention to practice and in pursuing, preparing, and filing the Patent Application. More specifically, on September 29, 1999, information similar to that shown in Exhibit A was presented to an evaluator to determine whether a patent application should be prepared, with the decision being made to proceed with preparing and filing a patent application.

[0022] Generally, the invention was conceived on or about March 1999 and was reduced to practice on September 1999. An exhaustive series of experiments were conducted on the invention from March 1999 to September 1999. The testing was quite rigorous and required substantial time, money, and effort to undertake. The results of the experiments were positive, which further resolved the decision to seek patent protection. After the invention disclosure form (Exhibit A) was submitted to the review and evaluation team within IBM, which is the customary business practice of IBM, the decision was reached to seek patent protection due to the potential commercial value and prestige afforded by the claimed invention. This decision was reached on or about October 7, 1999, and on October 14, 1999 a patent attorney was instructed to prepare a patent application that eventually became the Patent Application. The Patent Application was eventually prepared and filed on October 18, 2000. During the prosecution of the Patent Application, a Restriction Requirement was required by the USPTO. We elected to cancel claims 28-41 (as non-elected claims) of the Patent Application, and reserved the right to file a divisional patent application prior to issuance of the Patent Application. The divisional patent application was filed on September 11, 2003.

[0023] The forgoing declarations are made according to our best recollection upon review of the appropriate documents and notes, and we hereby acknowledge that willful false statements and the like are punishable by fine or imprisonment, or both (18 U.S.C. 1001) and may jeopardize the validity of the application or any patent issuing thereon. All statements made herein are made of our own knowledge and are true and all statements that are made on information and belief are believed to be true.


Dharmendra S. Modha

5/17/2006
Date


William Scott Spangler

5/18/2006
Date



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re patent application of:
Modha et al.

Serial No.: 10/660,242

Filed: September 11, 2003

Group Art Unit: 2165

Examiner: Mofiz, Apu M.

Atty. Docket No.: AM9990184US2

Certificate of Mailing

I hereby certify that this correspondence is being
mailed via Express Mail to the United States
Patent and Trademark Office and addressed to
Commissioner for Patents P.O. Box 1450
Alexandria, VA 22313-1450 on
May 23, 2006

Mohammad S. Rahman

For: CLUSTERING HYPERTEXT WITH APPLICATIONS TO WEB SEARCHING

AMENDMENT UNDER 37 C.F.R. §1.111

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This amendment is in response to the Office Action mailed April 10, 2006, setting a three-month statutory period for response. Therefore, this amendment is timely filed. Please amend the above-identified patent application as follows:

IN THE SPECIFICATION:

Please substitute the following paragraph in the application:

Page 1, after line 3:

Cross-Reference to Related Applications

This application is a division of U.S. Application Serial Number 09/690,854 filed October 18, 2000, the complete disclosure of which, in its entirety, is herein incorporated by reference.

IN THE ABSTRACT:

Please substitute the following abstract for the original abstract in the application:

~~A method and structure for providing a database of documents comprising performing a search of the database using a query to produce query result documents, constructing a word dictionary of words within the query result documents, pruning function words from the word dictionary, forming first vectors for words remaining in a word dictionary, constructing an out-link dictionary of documents within the database that are pointed to by the query result documents, adding the query result documents to the out link dictionary, pruning documents from the out-link dictionary that are pointed to by fewer than a first predetermined number of the query result documents, forming second vectors for documents remaining in the out-link dictionary, constructing an in-link dictionary of documents within the database that point to the query result documents, adding the query result documents to the in-link dictionary, pruning documents from the in-link dictionary that point to fewer than a second predetermined number of the query result documents, forming third vectors for documents remaining in the in-link dictionary, normalizing the first vectors, the second vectors, and the third vectors to create vector triplets for document remaining in the in-link dictionary and the out-link dictionary, clustering the vector triplets using the *toric k-means* process, and annotating/summarizing the obtained clusters using nuggets of information, the nuggets including summary, breakthrough, review, keyword, citation, and reference.~~

A method of searching a database of documents, wherein the method includes performing a search of the database using a query to produce query result documents; constructing a word dictionary of words within the query result documents; constructing an out-link dictionary of

documents within the database that are pointed to by the query result documents; adding the query result documents to the out-link dictionary; constructing an in-link dictionary of documents within the database that point to the query result documents; and adding the query result documents to the in-link dictionary.

IN THE CLAIMS:

Please substitute the following claims for the same-numbered claims in the application:

1-27. (Canceled).

28. (Currently Amended) A method of searching a database of documents, said method comprising:

performing a search of said database using a query to produce query result documents;
constructing a word dictionary of words within said query result documents;
constructing an out-link dictionary of documents within said database that are pointed to
by said query result documents; ~~and~~
adding said query result documents to said out-link dictionary;
constructing an in-link dictionary of documents within said database that point to said
query result documents; and
adding said query result documents to said in-link dictionary.

29. (Original) The method in claim 28, further comprising:

forming first vectors for words remaining in said word dictionary;
forming second vectors for documents remaining in said out-link dictionary;
forming third vectors for documents remaining in said in-link dictionary;
normalizing said first vectors, said second vectors, and said third vectors to create vector
triplets for document remaining in said in-link dictionary and said out-link dictionary; and

clustering the said vector triplets into one of clusters, or classes and partitions.

30. (Currently Amended) The method in claim 29, ~~where~~ wherein said clustering comprises a four *toric k-means* process comprising:

- (a) arbitrarily segregating the vector triplets into clusters;
- (b) for each cluster, computing a set of concept triplets describing said cluster;
- (c) re-segregating said vector triplets into a more coherent set of clusters by putting each vector triplet into a cluster corresponding to a concept triplet that is most similar to, a given vector triplet;
- (d) determining a coherence for each of said clusters based on a similarity of vector triplets within each of said clusters, and repeating the computing and re-segregating steps (b) (c) until coherence of the obtained clusters no longer significantly increases.

31. (Original) The method in claim 29, further comprising annotating and summarizing said clusters using nuggets of information, said nuggets including summary, breakthrough, review, keyword, citation, and reference.

32. (Original) The method in claim 31, wherein said summary comprises a document in a cluster having a most typical in-link feature vector amongst all the documents in said cluster.

33. (Original) The method in claim 31, wherein said breakthrough comprises a document in a cluster having a most typical in-link feature vector amongst all the documents in said cluster.

34. (Original) The method in claim 31, wherein said review comprises a document in a cluster having a most typical out-link feature vector amongst all the documents in said cluster.

35. (Original) The method in claim 31, wherein said keyword comprises a word in said word dictionary for said cluster that has the largest weight.

36. (Currently Amended) The method in claim 31, wherein said citation comprises a document in a cluster representing a most typical in-link into said cluster.

37. (Original) The method in claim 31, wherein said reference comprises a document in a cluster representing a most typical out-link out of said cluster.

38. (Original) The method in claim 28, further comprising pruning function words from said word dictionary.

39. (Original) The method in claim 28, further comprising pruning documents from said out-link dictionary that are pointed to by fewer than a first predetermined number of said query result documents.

40. (Original) The method in claim 28, further comprising pruning documents from said in-link dictionary that point to fewer than a second predetermined number of said query result documents.

41-54. (Canceled).

55. (New) A program storage device readable by computer, tangibly embodying a program of instructions executable by said computer to perform a method of searching a database of documents, said method comprising:

- performing a search of said database using a query to produce query result documents;
- constructing a word dictionary of words within said query result documents;
- constructing an out-link dictionary of documents within said database that are pointed to by said query result documents;
- adding said query result documents to said out-link dictionary;
- constructing an in-link dictionary of documents within said database that point to said query result documents; and
- adding said query result documents to said in-link dictionary.

56. (New) The program storage device in claim 55, wherein said method further comprises:

- forming first vectors for words remaining in said word dictionary;
- forming second vectors for documents remaining in said out-link dictionary;
- forming third vectors for documents remaining in said in-link dictionary;
- normalizing said first vectors, said second vectors, and said third vectors to create vector triplets for document remaining in said in-link dictionary and said out-link dictionary; and
- clustering the said vector triplets into one of clusters, or classes and partitions.

57. (New) The program storage device in claim 56, wherein in said method, said clustering

comprises a four *toric k-means* process comprising:

arbitrarily segregating the vector triplets into clusters;

for each cluster, computing a set of concept triplets describing said cluster;

re-segregating said vector triplets into a more coherent set of clusters by putting each vector triplet into a cluster corresponding to a concept triplet that is most similar to, a given vector triplet;

determining a coherence for each of said clusters based on a similarity of vector triplets within each of said clusters, and repeating the computing and re-segregating steps until coherence of the obtained clusters no longer significantly increases.

58. (New) The program storage device in claim 56, wherein said method further comprises annotating and summarizing said clusters using nuggets of information, said nuggets including summary, breakthrough, review, keyword, citation, and reference.

59. (New) The program storage device in claim 58, wherein said summary comprises a document in a cluster having a most typical in-link feature vector amongst all the documents in said cluster, wherein said breakthrough comprises a document in a cluster having a most typical in-link feature vector amongst all the documents in said cluster, wherein said review comprises a document in a cluster having a most typical out-link feature vector amongst all the documents in said cluster, wherein said keyword comprises a word in said word dictionary for said cluster that has the largest weight, wherein said citation comprises a document in a cluster representing a most typical in-link into said cluster, and wherein said reference comprises a document in a cluster representing a most typical out-link out of said cluster.

60. (New) The program storage device in claim 55, wherein said method further comprises:

pruning function words from said word dictionary;

pruning documents from said out-link dictionary that are pointed to by fewer than a first predetermined number of said query result documents; and

pruning documents from said in-link dictionary that point to fewer than a second predetermined number of said query result documents.

61. (New) A system for searching a database of documents, said system comprising:

means for performing a search of said database using a query to produce query result documents;

means for constructing a word dictionary of words within said query result documents;

means for constructing an out-link dictionary of documents within said database that are pointed to by said query result documents;

means for adding said query result documents to said out-link dictionary;

means for constructing an in-link dictionary of documents within said database that point to said query result documents; and

means for adding said query result documents to said in-link dictionary.

REMARKS

Claims 28-40 and 55-61 are all the claims pending in the application. The Office Action indicates that claims 1-54 stand rejected under the judicially created doctrine of obviousness-type double patenting and on prior art grounds. However, claims 1-27 and 42-54 were previously canceled in the Preliminary Amendment filed on September 11, 2003. Additionally, claim 41 is cancelled herein without prejudice or disclaimer. Moreover, claims 28, 30, and 36 are amended herein and claims 55-61 are newly added. Furthermore, the specification and abstract are amended for further clarification of the Applicants' invention. No new matter is being added. Applicants respectfully traverse these rejections based on the following discussion.

I. The Double Patenting Rejection

Claims 1-54 stand rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1-24 of U.S. Patent No. 6,684,205 and claims 1-8 of U.S. Patent No. 6,862,586. Applicants respectfully traverse these rejections based on the following discussion.

U.S. Patent No. 6,684,205 is the parent application to the present divisional application. Accordingly to MPEP §§804.01, 806, and 806.05 it is improper to issue a double patenting rejection in the case where the rejected claims were originally presented in a parent application and were the subject of a mandatory restriction requirement in the parent case thereby resulting in non-elected claims, which subsequently become the subject of a divisional application. In the current situation, in the prosecution of U.S. Patent No. 6,684,205, a Restriction Requirement was issued by the USPTO on October 23, 2002 because the Restriction Requirement stated that the application contained two distinct inventions (termed Invention I (claims 1-27 and 42-54) and

Invention II (claims 28-41) in the Restriction Requirement). The Applicants responded to the Restriction Requirement on October 31, 2002 by electing Invention I (claims 1-27 and 42-54) without traverse. Subsequently claims 28-41 (non-elected claims) were cancelled without prejudice or disclaimer in an Amendment filed on February 28, 2003. Thereafter, the present application was filed on September 11, 2003 as a divisional application, which was prior to issuance of the parent application. Furthermore, while the claims in the parent application were amended during prosecution of the parent application, the amended claimed language did not overlap with the claimed language in the present application (divisional application). Accordingly, because the Restriction Requirement was required by the USPTO initially, the aforementioned sections of the MPEP clearly indicate that it is improper to issue a double patenting rejection of the non-elected claims in the parent application subsequently set forth in a divisional application. In view of the foregoing, the Examiner is respectfully requested to reconsider and withdraw this rejection.

With respect to U.S. Patent No. 6,862,586, the Office Action states that the claims of U.S. Patent No. 6,862,586 contain every element of claims 1-54 of the instant specification. However, it should be noted that claims 1-27 and 42-54 of the present application have been previously cancelled and claim 41 is cancelled herewith without prejudice or disclaimer. Accordingly, the following remarks are limited to claims 28-40 (newly added claims 55-61 are program storage claims and system claims related to the same subject matter recited in claims 28-40).

U.S. Patent No. 6,862,586 provides the following claims 1-8:

1. A method of perforating a database search comprising: searching a database using a query, said searching identifying a group of hyperlinked documents; forming a high-dimensional torus geometric representation of said hyperlinked

documents, wherein each hyperlinked document is represented by a vector triplet comprising a normalized word frequency, a normalized out-link frequency and a normalized in-link frequency; clustering said result items into clusters based on said high-dimensional torus geometric representation; ranking items within each cluster of said clusters based on said high-dimensional torus geometric representation; summarizing contents of said clusters based on said high-dimensional torus geometric representation, wherein said clustering of the said vector triplets on said high-dimensional torus geometric representation is performed using a toric k-means clustering process that uses a cosine-type similarity measure between document vector triplets, thereby producing clusters of vector triplets and producing a concept triplet for each of the clusters; and summarizing said clusters of vector triplets based on nuggets of information including: identifying a closeness of said vector triplets in a cluster to said concept triplet for said cluster on said high-dimensional torus geometric representation; identifying said words with a highest normalized word frequency in said concept triplet for said cluster as the most frequent key-words for each of said clusters; identifying said out-links with a highest normalized out-link frequency in the concept triplet for the cluster as most frequent key out-links for each of said clusters; identifying said in-links with a highest normalized in-link frequency in the concept triplet for the cluster as most frequent important in-links for each cluster; identifying hypertext items relevant to the user's query by using a weighting of terms used in said query; identifying documents closest to said concept triplet as most typical documents in a cluster, using a cosine-type textual content similarity measure between document vector triplets; and identifying documents closest to said concept triplet as most typical documents in a cluster, using a cosine-type out-link similarity measure between document vector triplets; and identifying documents closest to said concept triplet as most typical documents in a cluster, using a cosine-type in-link similarity measure between document vector triplets.

2. A method of performing a database search comprising: searching a database using a query, said searching identifying a group of documents; forming a high-dimensional torus geometric representation of said documents, wherein each document is represented by a vector triplet comprising a normalized word frequency, a normalized out-link frequency and a normalized in-link frequency; identifying documents closest to a concept triplet as most typical documents in a cluster, using a cosine-type out-link similarity measure between document vector triplets; and identifying documents closest to said concept triplet as most typical documents in a cluster, using a cosine-type in-link similarity measure between document vector triplets.

3. The method in claim 2, further comprising: clustering said result items into clusters based on said high-dimensional torus geometric representation; ranking items within each cluster of said clusters based on said high-dimensional torus geometric representation; and summarizing contents of said clusters based on said

high-dimensional torus geometric representation.

4. The method in claim 3, wherein said clustering comprises agglomerative clustering, hierarchical clustering, EM algorithm, or mixture modeling.
5. The method in claim 3, wherein said ranking includes identifying a most typical vector triplet in each of said clusters of vector triplets.
6. The method in claim 2, wherein said normalized out-link frequency comprises a number of said documents linked to, cited, or pointed to by said document.
7. The method in claim 2, wherein said normalized in-link frequency comprises a number of said documents linking to, citing, or pointing to said document.
8. The method in claim 2, wherein said normalized word frequency comprises a number of unique words, terms, or n-grams contained in said document.

However, nowhere in the above claims is there any language pertaining to “constructing a word dictionary of words within said query result documents; constructing an out-link dictionary of documents within said database that are pointed to by said query result documents; adding said query result documents to said out-link dictionary; constructing an in-link dictionary of documents within said database that point to said query result documents; and adding said query result documents to said in-link dictionary” as generally provided in the Applicants’ broadest independent claims 28, 55, and 61. Additionally, there is no teaching or suggestion in claims 1-8 of U.S. Patent No. 6,862,586 of a “four *toric k-means* process” or “annotating and summarizing said clusters using nuggets of information, said nuggets including summary, breakthrough, review, keyword, citation, and reference” or “pruning documents from said in-link dictionary that point to fewer than a second predetermined number of said query result documents.”

Accordingly, claims 1-8 of U.S. Patent No. 6,862,586 and the Applicants’ present claims 28-40 and 55-61 are patentably distinct because claims 1-8 of U.S. Patent No. 6,862,586 do not contain every element of claims 28-40 and 55-61 of the present application. In view of the foregoing,

the Examiner is respectfully requested to reconsider and withdraw this rejection.

II. The Prior Art Rejections

Claims 1-6, 15-19, 28, 38-40, and 42-46 stand rejected under 35 U.S.C. §102(a) as being anticipated by Kuo et al. (Web Document Classification based on Hyperlinks and Document Semantics, August 2000, pp.44-51), hereinafter referred to as “Kuo”. Applicants respectfully traverse these rejections based on the following discussion. Claims 6-14, 19-27, and 46-54 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Kuo in view of Pirolli et al. (Silk from Sow’s Ear: Extracting Usable Structures from the Web, CHI 1996, pages 1-9), hereinafter referred to as “Pirolli”. Applicants respectfully traverse these rejections based on the following discussion.

It should be noted that claims 1-27 and 42-54 of the present application have been previously cancelled and claim 41 is cancelled herewith without prejudice or disclaimer. Therefore, the rejection under 35 U.S.C. §103(a) is immaterial. Accordingly, the following remarks are limited to the rejection of claims 28-40 (newly added claims 55-61 are program storage claims and system claims related to the same subject matter recited in claims 28-40) under 35 U.S.C. §102(a) as being anticipated by Kuo.

Kuo teaches a web document that also contains a set of hyperlinks pointing to other related documents. Hyperlinks in a document provide much information about its relation with other web documents. By analyzing hyperlinks in documents, inter-relationship among documents can be identified. Kuo provides an algorithm to classify web documents into subsets based on hyperlinks in documents and their content. Representative documents are identified in each subset based on a similarity definition. With the representative document, searching for

related documents can be achieved.

The Kuo reference is relied upon as teaching the entirety of claims 28 and 38-40. Submitted herewith is a Rule 131 Declaration swearing behind the Kuo reference. The Rule 131 Declaration removes the Kuo reference from consideration. In view of the foregoing, the Examiner is respectfully requested to reconsider and withdraw this rejection.

III. Entry of Amendment and Rule 131 Declaration Required

MPEP § 715.09 provides that a Rule 131 Declaration is considered timely submitted if it is submitted prior to a final rejection. Therefore, the attached Rule 131 Declaration swearing behind the Kuo reference is seasonably presented.

IV. Formal Matters and Conclusion

With respect to the rejections to the claims, the Applicants respectfully traverse the double patenting rejection. Additionally, the Rule 131 Declaration that accompanies this Amendment swears behind the Kuo reference thereby overcoming the prior art rejections. In view of the foregoing, the Examiner is respectfully requested to reconsider and withdraw the rejections to the claims.

In view of the foregoing, the Applicants submit that claims 28-40 and newly added claims 55-61, all the claims presently pending in the application, are patentably distinct from the prior art of record and are in condition for allowance. Furthermore, no new matter is being presented. The Examiner is respectfully requested to pass the above application to issue at the earliest possible time.

Should the Examiner find the application to be other than in condition for allowance, the

Examiner is requested to contact the undersigned at the local telephone number listed below to discuss any other changes deemed necessary. Please charge any deficiencies and credit any overpayments to Attorney's Deposit Account Number 09-0441.


Respectfully submitted,

Dated: May 23, 2006

A handwritten signature in black ink, appearing to read 'Mohammad S. Rahman', written over a horizontal line.

Mohammad S. Rahman
Registration No. 43,029

Gibb I.P. Law Firm, LLC
2568-A Riva Road, Suite 304
Annapolis, MD 21401
Voice: (301) 261-8625
Fax: (301) 261-8825
Customer Number: 29154

	Draft Disclosure	8990272
	Created By: Dharmendra Modha	Created On: 05:05:31 PM
	Last Modified By: wpts1 wpts1	Last Modified On: 11:54:43 AM
*** IBM Confidential ***		

Summary

Required fields are marked with the asterisk (*) and must be filled in to complete the form.

Status	Submitted
Processing Location	ARC
Functional Area	DPB - Computer Science - (A.K. Chandra)
Submitted Date	05:17:29 PM
Owning Division	
PVT Score	To calculate a PVT score, use the 'Calculate PVT' button.
Lab	
Technology Code	
Incentive Program	

Inventors with Lotus Notes IDs

Inventors: Dharmendra Modha/Almaden/IBM, W Spangler/Almaden/IBM

Inventor Name > denotes primary contact	Inventor Serial	Div/Dept	Manager Serial	Manager Name
> Modha, Dharmendra	042675	22/K53F	984805	Strong, H. Raymond (Ray)
Spangler, W. Scott	894199	22/K57F	002078	Kreulen, Jeffrey T.

Inventors without Lotus Notes IDsIDT Selection *Mark McSwain*

IDT Team: <i>Rich Indurain</i>	Attorney/Patent Professional:
--------------------------------	-------------------------------

Main Idea***Title of disclosure (in English)**

Clustering Hypertext with Applications to Web Searching

***Idea of disclosure**

1. Describe your invention, stating the problem solved (if appropriate), and indicating the advantages of using the invention.

Clustering separates unrelated documents and groups related documents, and is useful for discrimination, disambiguation, summarization, organization, and navigation of unstructured collections of hypertext documents. We propose a novel clustering algorithm that clusters hypertext documents using words (contained in the document), out-links (from the document), and in-links (to the document). The algorithm automatically determines the relative importance of words, out-links, and in-links for a given collection of hypertext documents. We annotate each cluster using six information nuggets: summary, breakthrough, review, keywords, citation, and reference. These nuggets constitute high-quality information resources, and are extremely effective in compactly summarizing and navigating the collection of hypertext documents. We employ web searching as an application to illustrate our results.

Clustering Hypertext with Applications to Web Searching

Dharmendra S. Modha and W. Scott Spangler

IBM Almaden Research Center

650 Harry Road, San Jose, CA 95 120-6099

email: {dmodha, spangles}@almaden.ibm.com

ABSTRACT

Clustering separates unrelated documents and groups related documents, and is useful for discrimination, disambiguation, summarization, organization, and navigation of unstructured collections of hypertext documents. We propose a novel clustering algorithm that clusters hypertext documents using words (contained in the document), out-links (from the document), and in-links (to the document). The algorithm automatically determines the relative importance of words, out-links, and in-links for a given collection of hypertext documents. We annotate each cluster using six information nuggets: *summary*, *breakthrough*, *review*, *keywords*, *citation*, and *reference*. These nuggets constitute high-quality information resources that are representatives of the content of the clusters, and are extremely effective in compactly summarizing and navigating the collection of hypertext documents. We employ web searching as an application to illustrate our results.

KEYWORDS: cluster annotation, feature combination, high-dimensional data, hyperlinks, sparse data, vector space model, toric k-means algorithm

INTRODUCTION

The World-Wide-Web has attained a gargantuan size [16] and a central place in the information economy of today. Hypertext is the *lingua franca* of the web. Moreover, scientific literature, patents, and law cases may be thought of as logically hyperlinked. Consequently, searching and organizing unstructured collections of hypertext documents is a major contemporary scientific and technological challenge.

Given a “broad-topic query” [13], a typical web search engine may return a large number of relevant (and irrelevant) documents. Without effective summarization, it is a hopeless and enervating task to sort through all the returned documents in search of high-quality, representative information resources. In this paper, we cluster the set of hypertext documents that are returned by a search engine in response to a

broad-topic query into various clusters such that documents within each cluster are “similar” to each other. Clustering provides a way to organize a large collection of unstructured, unlabeled hypertext documents into labeled categories that are discriminated and disambiguated from each other. Furthermore, we capture the gist of each cluster using a scheme for cluster annotation that provides useful starting points for navigating/surfing in and around each cluster.

Ignoring the semantic information present in various HTML tags, a hypertext document has three different features: (i) the words contained in the document, (ii) out-links, that is, the list of hypertext documents that are *pointed to* or *cited* by the document, and (iii) the in-links, that is, the list of hypertext documents that *point to* or *cite* the document. We exploit all the three features to cluster a collection of hypertext documents. If two documents share one or more words, then we consider them to be semantically similar. Extending this notion to links, if two documents share one or more out-links or in-links, then we consider them to be similar as well. This simple observation is the key to the present paper. We propose a precise notion to capture the similarity between two hypertext documents along all the three features in an unified fashion. By exploiting our new similarity measure, we propose a geometric hypertext clustering algorithm: *the toric k-means* that extends the classical Euclidean k-means algorithm [12] and the spherical k-means algorithm [9, 19].

We annotate each cluster generated by the toric k-means algorithm using six information nuggets: *summary*, *breakthrough*, *review*, *keywords*, *citation*, and *reference*. The *summary* and the *keywords* are derived from words, the *review* and the *references* are derived from out-links, and the *breakthrough* and the *citations* are derived from in-links. These nuggets constitute high-quality, typical information resources, and are extremely effective in compactly summarizing and navigating the collection of hypertext documents.

The relative importances of the words, the out-links, and the in-links are tunable parameters in our algorithm. We propose an *adaptive* or *data-driven* scheme to determine these parameters with the goal of simultaneously improving the quality of all the six information nuggets for all the clusters.

Throughout the paper, we employ web searching as an application to illustrate our results. Anecdotally, when applied

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Hypertext 2000, San Antonio, TX.

Copyright 2000 ACM 1-581 13-227-1/00/0006...\$5.00

to the documents returned by AltaVista in responses to the queries *latex*, *abduction*, *guinea*, and *abortion*, our algorithm separates documents about “latex allergies” from those about “ \TeX & \LaTeX ,” separates documents about “alien abduction” from those about “child abduction,” separates documents about “Papua New Guinea,” “Guinea Bissau,” and “Guinea pigs” from each other, and separates documents about “pro-life” from those about “pro-choice”, respectively.

We include directions for future work and a detailed literature survey at the end of the paper.

A GEOMETRIC ENCODING OF THE WEB

The Data Set Suppose we are given a collection of hypertext documents, say, \mathcal{W} . Let \mathcal{Q} denote a subset of \mathcal{W} . In this paper, for example, \mathcal{W} denotes the entire web, and \mathcal{Q} denotes a small collection of hypertext documents retrieved by the search engine AltaVista (www.altavista.com) in response to a query. We are interested in clustering the hypertext documents in \mathcal{Q} . The situation of interest is depicted in Figure 1, where we have only shown those documents that are at most one out- or in-link away from the documents in \mathcal{Q} ; in this paper, all other link information is discarded. The words contained in hypertext documents are not shown in Figure 1.

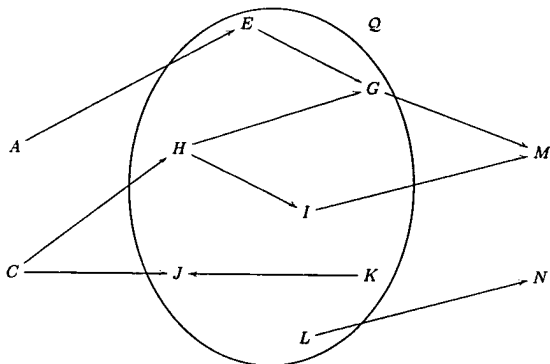


Figure 1: We are interested in clustering the set $\mathcal{Q} = \{E, G, H, I, J, K, L\}$ of hypertext documents. The documents $\{A, C, M, N\}$ are not in \mathcal{Q} , but are hyperlinked to the documents in \mathcal{Q} .

We now extract useful features from \mathcal{Q} and propose a geometric representation for these features. We will represent each hypertext document in \mathcal{Q} as a triplet of unit vectors (D, F, B) . These component vectors are to be thought of as column vectors. The components D , F , and B will capture the information represented by the words contained in the document, the out-links originating at the document, and the in-links terminating at the document, respectively. We now show how to compute these triplets for each document in \mathcal{Q} .

Words The creation of the first component D is a standard exercise in text mining or information retrieval, see [21].

The basic idea is to construct a *word dictionary* of all the

words that appear in any of the documents in \mathcal{Q} , and to prune or eliminate “function” words from this dictionary that do not help in semantically discriminating one cluster from another. For the present application, we eliminated those words which appeared in less than 2 documents, standard stopwords [10], and the HTML tags.

Suppose d unique words remain in the dictionary after such elimination. Assign an unique identifier from 1 to d to each of these words. Now, for each document x in \mathcal{Q} , the first vector D in the triplet will be a d -dimensional vector. The j th column entry, $1 \leq j \leq d$, of D is the number of occurrences of the j th word in the document x .

Out-links We now outline the creation of the second component F . The basic idea is to construct an *out-link dictionary* of all the hypertext documents in $\mathcal{W} \setminus \mathcal{Q}$ that are pointed to by any of the documents in \mathcal{Q} . We also add each document in \mathcal{Q} to the out-link dictionary. For example, in Figure 1, the out-link dictionary is $\{E, G, H, I, J, K, L, M, N\}$.

To treat nodes in $\mathcal{W} \setminus \mathcal{Q}$ and in \mathcal{Q} in a uniform fashion, we add a self-loop from every document in \mathcal{Q} to itself. Any document in the out-link dictionary that is not pointed to by at least two documents in \mathcal{Q} provides no *discriminating* information. Hence, prune or eliminate all documents from the out-link dictionary that are pointed to by fewer than two documents (also counting the self-loops) in \mathcal{Q} . For example, in Figure 1, we eliminate the node N as it is pointed to by only L , but retain M as it is pointed to by both G and I . Similarly, we eliminate the nodes E, H, K , and L as they are not pointed to by any document in \mathcal{Q} other than themselves, but retain G, I , and J as they are pointed to by at least one other document in \mathcal{Q} and by themselves.

Suppose f unique nodes remain in the dictionary after such elimination. Assign an unique identifier from 1 to f to each of these documents. Now, for each document x in \mathcal{Q} , the second vector F in the triplet will be a f -dimensional vector. The j th column entry, $1 \leq j \leq f$, of F is the number of links to the j th retained node from the document x . We now present the out-link feature vectors for the example in Figure 1:

	E	G	H	I	J	K	L
G	1	1	1	0	0	0	0
I	0	0	1	1	0	0	0
J	0	0	0	0	1	1	0
M	0	1	0	1	0	0	0

In-links The creation of B is similar to that of F ; for completeness, we now briefly describe its construction. The basic idea is to construct an *in-link dictionary* of all the hypertext documents in $\mathcal{W} \setminus \mathcal{Q}$ that point to any of the documents in \mathcal{Q} . We also add each document in \mathcal{Q} to the in-link dictionary.

To treat nodes in $\mathcal{W} \setminus \mathcal{Q}$ and in \mathcal{Q} in a uniform fashion, we add a self-loop from every document in \mathcal{Q} to itself. Any doc-

ument in the in-link dictionary that does not point to at least two documents in \mathcal{Q} provides no *discriminating* information. Hence, prune or eliminate all documents from the in-link dictionary that point to fewer than two documents (also counting the self-loops) in \mathcal{Q} .

Suppose b unique nodes remain in the dictionary after such elimination. Assign an unique identifier from 1 to b to each of these documents. Now, for each document x in \mathcal{Q} , the third vector B in the triplet will be a b -dimensional vector. The j th column entry, $1 \leq j \leq b$, of B is the number of links from the j th retained node to the document x .

Normalization Finally, for each document x in \mathcal{Q} , each of the three components D , F , and B is normalized to have a unit Euclidean norm, that is, their directions are retained and their lengths are discarded.

Torus We now briefly point out the geometry underlying our three-fold vector space models. Suppose that we have n documents in \mathcal{Q} . We denote each document triplet as

$$x_i = (D_i, F_i, B_i), 1 \leq i \leq n.$$

Observe that, by construction, the component vectors D_i , F_i , and B_i all have unit Euclidean norm, and, hence, can be thought of as points on the unit spheres S^d , S^f , and S^b in dimensions d , f , and b , respectively. Thus, each document triplet x_i lies on the product space of three spheres, which is a *torus*, see (www.treasure-troves.com/math/Torus.html). Furthermore, by construction, the individual entries of the component vectors D_i , F_i , and B_i are nonnegative, hence, the component vectors are in fact in the nonnegative orthants of R^d , R^f , and R^b , respectively. For notational convenience, we refer to the intersection of $(S^d \times S^f \times S^b)$ with the non-negative orthant of R^{d+f+b} as T .

AltaVista: Details Given a user query, we run it through AltaVista which typically returns a list of 200 URLs containing the keywords in the query. We crawl, and retrieve each of these 200 documents (those documents that could not be retrieved in 1 minute were discarded), and that becomes our set \mathcal{Q} . Next, we parse each of these documents, and construct the unpruned out-link dictionary. Finally, for each document in \mathcal{Q} , using queries of the form “link:URL” on AltaVista, we retrieve the URLs of top 20 documents that point to it. This constitutes our unpruned in-link dictionary. Observe that we do not need the actual documents in either the out- or the in-link dictionary. The set \mathcal{Q} and the out- and the in-link dictionaries now become the inputs for the vector space model construction procedure described above.

Statistics In Table 1, for a number of queries, we present statistical properties of the three-fold vector space models.

High-dimensional By observing the d , f , and b values in Table 1, we see that, even after pruning, the word, out-link, and in-link dictionaries are very high-dimensional. Also, typically, d is the much larger than both f and b .

Sparse By observing the ratios d°/d , f°/f , and b°/b in Table 1, we see that the vector space models are very sparse. A sparsity of 96% is typical for words, that is, on an average each document contains only 4% of the words in the word dictionary. Similarly, sparsities of 95–98% and 91–97% are typical for out- and in-links, respectively.

By observing the n_f and n_b values, we see that not all documents have nonzero out-link and in-link features vectors. This points once again to the sparse link topology that is holding the web together. Also, the variations in n_f and n_b values point to the fact that some “topics” or “communities” are more tightly coupled than others.

Importance of $\mathcal{W} \setminus \mathcal{Q}$ Finally, by observing the \hat{n}_f and \hat{n}_b values, we see that the number of nodes from the original set \mathcal{Q} retained in the final pruned out-link and in-link dictionaries is rather small. In other words, the interconnection structure of the set \mathcal{Q} with the rest of the web, namely, $\mathcal{W} \setminus \mathcal{Q}$, contains the vast majority of the link information in our feature vectors. This justifies our decision to include links between the documents in \mathcal{Q} and the documents $\mathcal{W} \setminus \mathcal{Q}$.

TORIC k -MEANS ALGORITHM

A Measure of Similarity Given document triplets $x = (D, F, B)$ and $\tilde{x} = (\tilde{D}, \tilde{F}, \tilde{B})$ on the torus T , we define a measure of similarity between them as a weighted sum of the inner products between the individual components. Precisely, we write

$$S(x, \tilde{x}) = \alpha_d D^T \tilde{D} + \alpha_f F^T \tilde{F} + \alpha_b B^T \tilde{B}, \quad (1)$$

where *weights* α_d , α_f , and α_b are nonnegative numbers such that

$$\alpha_d + \alpha_f + \alpha_b = 1.$$

Observe that for any two document triplets x and \tilde{x} , $0 \leq S(x, \tilde{x}) \leq 1$. Also, observe that if we set $\alpha_d = 1$, $\alpha_f = 0$, and $\alpha_b = 0$, then we get the classical cosine similarity between document vectors that has been widely used in information retrieval [21]. The parameters α_d , α_f , and α_b are tunable in our algorithm to assign different weights to words, outlinks, and in-links as desired. We will later discuss, in detail, the appropriate choice of these parameters.

Concept Triplets Suppose we are given n document vector triplets x_1, x_2, \dots, x_n on the torus T . Let $\pi_1, \pi_2, \dots, \pi_k$ denote a partitioning of these document triples into k *disjoint* clusters. For each fixed $1 \leq j \leq k$, the *concept vector triplet* or *concept triplet*, for short, is defined as

$$c_j = (D_j^*, F_j^*, B_j^*) \quad (2)$$

$$D_j^* = \frac{x \in \pi_j D}{\|x \in \pi_j D\|}, F_j^* = \frac{x \in \pi_j F}{\|x \in \pi_j F\|}, B_j^* = \frac{x \in \pi_j B}{\|x \in \pi_j B\|}. \quad (3)$$

where $x = (D, F, B)$ and $\|\cdot\|$ denotes the Euclidean norm. Observe that, by construction, each component of the concept triplet has unit Euclidean norm. The concept triplet c_j

query	n	d°	d	d°	n_d	f°	f	f°	n_f	\hat{n}_f	b°	b	b°	n_b	\hat{n}_b
latex	148	7059	2706	100.3	148	922	92	3.2	55	11	585	28	1.4	45	7
abortion	156	6205	2670	101.6	156	1286	144	3.3	67	10	662	58	1.8	64	9
guinea	146	6600	2814	100.0	146	1392	351	17.8	67	12	585	54	1.9	70	6
abduction	155	5967	2643	97.2	155	677	81	3.2	46	9	378	38	2.6	40	6
virus	146	6118	2627	111.6	146	2601	765	20.5	95	18	1191	100	3.8	70	14
"human rights"	157	7314	2800	113.6	157	1446	204	4.6	77	14	1369	99	2.9	71	12
dilbert	164	4584	1934	74.8	164	1257	173	3.8	80	4	385	15	1.3	45	5
terrorism	154	9824	4493	208.5	154	1762	242	6.1	74	18	675	47	2.0	51	14

Table 1: A note on notation: n represents the number of documents in \mathcal{Q} , d° and d are the number of words in the word-dictionary before and after elimination of function words, respectively, d° is the average number of nonzero word counts per document, and n_d is the number of documents which contain at least one word after elimination. The symbols f° , f , f° , and n_f as well as the symbols b° , b , b° , and n_b have a similar meaning to their counterparts for the words. The symbols \hat{n}_f and \hat{n}_b are the number of documents in \mathcal{Q} that are eventually retained in the final, pruned out-link and the in-link dictionaries, respectively.

has the following important property. For any triplet $\tilde{x} = (\tilde{D}, \tilde{F}, \tilde{B})$ on the torus T , we have from the Cauchy-Schwarz inequality that

$$\sum_{x \in \pi_j} S(x, \tilde{x}) \leq \sum_{x \in \pi_j} S(x, c_j). \quad (4)$$

Thus, in an average sense, the concept triplet may be thought of as being the closest in S to all the document vector triplets in the cluster π_j .

We shall demonstrate that concept triplets contain valuable conceptual or semantic information about the clusters that is important in interpretation and annotation.

The Objective Function Motivated by (4), we measure the "coherence" or "quality" of each cluster π_j , $1 \leq j \leq k$, as

$$\sum_{x \in \pi_j} S(x, c_j).$$

Observe that if all documents in a cluster are identical, then the average coherence of that cluster will have the highest possible value of 1, while if the document vectors in a cluster vary widely, then the average coherence will be small, that is, close to 0. We measure the quality of any given partitioning $\{\pi_j\}_{j=1}^k$ using the following *objective function*:

$$\sum_{j=1}^k \sum_{x \in \pi_j} S(x, c_j). \quad (5)$$

Intuitively, the objective function measures the combined coherence of all the k clusters.

The Algorithm We would like to find k disjoint clusters $\pi_1^\dagger, \pi_2^\dagger, \dots, \pi_k^\dagger$ such that the following is maximized:

$$\{\pi_j^\dagger\}_{j=1}^k = \arg \max_{\{\pi_j\}_{j=1}^k} \left(\sum_{j=1}^k \sum_{x \in \pi_j} S(x, c_j) \right). \quad (6)$$

Even when only one of the parameters α_d , α_f , or α_b is nonzero, finding the optimal solution to the above maximization problem is known to be NP-complete. We now discuss an efficient and effective approximation algorithm: the *toric k-means* that may be thought of as a *gradient ascent* method.

Step 1 Start with an arbitrary partitioning of the document vectors, namely, $\{\pi_j^{(0)}\}_{j=1}^k$. Let $\{c_j^{(0)}\}_{j=1}^k$ denote the concept triplets associated with the given partitioning. Set the index of iteration $t = 0$. The choice of the initial partitioning is quite crucial to finding a "good" local minima; for recent work on this area, see [2].

Step 2 For each document vector triplet x_i , $1 \leq i \leq n$, find the concept triplet that is closest to x_i . Now, for $1 \leq j \leq k$, compute the new partitioning $\{\pi_j^{(t+1)}\}_{j=1}^k$ induced by the old concept triplets $\{c_j^{(t)}\}_{j=1}^k$:

$$\pi_j^{(t+1)} = \left\{ x \in \{x_i\}_{i=1}^n : S(x, c_j^{(t)}) \geq S(x, c_\ell^{(t)}), 1 \leq \ell \leq k \right\}. \quad (7)$$

In words, $\pi_j^{(t+1)}$ is the set of all document vector triplets that are closest to the concept triplet $c_j^{(t)}$. If it happens that some document triplet is simultaneously closest to more than one concept triplet, then it is randomly assigned to one of the clusters.

Step 3 Compute the new concept triplets $\{c_j^{(t+1)}\}_{j=1}^k$ corresponding to the partitioning computed in (7) by using (2)-(3) where instead of π_j we use $\pi_j^{(t+1)}$.

Step 4 If some "stopping criterion" is met, then set $\pi_j^\dagger = \pi_j^{(t+1)}$ and set $c_j^\dagger = c_j^{(t+1)}$ for $1 \leq j \leq k$, and exit. Otherwise, increment t by 1, and go to step 2 above. An example of a stopping criterion is: Stop if the change in the objective function, between two successive iterations, is less than some specified threshold.

Shape of Clusters Clusters defined using (7) are known as *Voronoi* or *Dirichlet* partitions. The boundary between two clusters, say, π_j^\dagger and π_ℓ^\dagger , is the locus of all document triplets x on T satisfying:

$$S(x, c_j^\dagger) = S(x, c_\ell^\dagger).$$

If only one of the parameters α_d , α_f , or α_b is nonzero, then the above locus is a hypercircle on the corresponding sphere; when more than one parameters is nonzero, the locus is a hyperellipsoid. Thus, each cluster is a region on the surface of the underlying torus bounded by hyperellipsoids. In conclusion, the geometry of the torus plays an integral role in determining the “shape” and the “structure” of the clusters found by the toric k -means algorithm.

CLUSTER ANNOTATION AND INTERPRETATION

Suppose that we have clustered a hypertext collection Q into k clusters $\{\pi_j^\dagger\}_{j=1}^k$; let $\{c_j^\dagger\}_{j=1}^k$ denote the corresponding concept triplets. In this raw form, the clustering is of little use. We now use the concept triplets to interpret and annotate each cluster. *The process of seeking good cluster annotation will motivate the choice of the weights α_d , α_f , and α_b .*

Fix a cluster π_j^\dagger , $1 \leq j \leq k$. Let $c_j^\dagger = (D_j^*, F_j^*, B_j^*)$ denote the corresponding concept triplet. We now show how to label the fixed cluster π_j^\dagger using six different nuggets of information whose names have been inspired by their respective analogues in the scientific literature.

summary A *summary* is a document in π_j^\dagger that has the most typical word feature vector amongst all the documents in the cluster. Formally, the *summary* is a document triplet $x = (D, F, B)$ whose word component D is closest in cosine similarity to D_j^* .

breakthrough A *breakthrough* is a document in π_j^\dagger that has the most typical in-link feature vector amongst all the documents in the cluster. Formally, the *breakthrough* is a document triplet $x = (D, F, B)$ whose in-link component B is closest in cosine similarity to B_j^* .

review A *review* is a document in π_j^\dagger that has the most typical out-link feature vector amongst all the documents in the cluster. Formally, the *review* is a document triplet $x = (D, F, B)$ whose out-link component F is closest in cosine similarity to F_j^* .

keywords *Keywords* for the cluster π_j^\dagger are those words in the word dictionary that have the largest weight in D_j^* compared to their respective weights in D_ℓ^* , $1 \leq \ell \leq k, \ell \neq j$. *Keywords* are the most discriminating words in a cluster, and constitute an easy-to-interpret cluster signature.

citations *Citations* for the cluster π_j^\dagger are those in-links in the in-link dictionary that have the largest weight in B_j^* compared to their respective weights in B_ℓ^* , $1 \leq \ell \leq k, \ell \neq j$.

Citations represent the set of most typical links *entering* (the documents in) the given cluster.

references *References* for the cluster π_j^\dagger are those out-links in the out-link dictionary that have the largest weight in F_j^* compared to their respective weights in F_ℓ^* , $1 \leq \ell \leq k, \ell \neq j$. *References* represent the set of most typical links *exiting* (from the documents in) the given cluster.

If we were interested in clustering a collection of not-hyperlinked text documents, then the *summary* and the *keywords* would constitute an adequate annotation. For hypertext collections, our annotation naturally extends the concepts of *summary* and the *keywords* from words to in-links and out-links as well. Observe that the *summary*, the *breakthrough*, and the *review* are meant to be primarily *descriptive* of the contents of the cluster, whereas the *keywords*, the *references*, and the *citations* are meant to be *discriminative* characteristics of the cluster. Also, observe that the *summary*, the *breakthrough*, and the *review* are, by definition, drawn from the set Q ; however, the *citations* and the *references* may or may not be in the set Q .

Effectiveness of Annotation: Examples Suppose, for a moment, that we are not interested in clustering at all; in other words, suppose that we are interested in only one cluster, that is, $k = 1$. Even in this case, the six nuggets described above are meaningful, and often capture the top information resources present in Q .

For example, in Table 2, by treating the entire set Q as one cluster, we present the six nuggets for each of the four queries: *virus*, “*human rights*,” *dilbert*, and *terrorism*. As even a casual glance reveals, the annotation indeed captures the top information resources in every case, and provides a valuable starting point for navigating the documents surrounding the cluster.

Furthermore, note that, in Table 2, every document that is in Q is followed by a parenthetic number that represents its rank in the documents returned by AltaVista. For example, for the query *virus* the *summary* is “Anti-Virus Tools (51)” meaning that it was the fifty-first document returned by AltaVista. By observing these parenthetic numbers, we can conclude that, in almost every case, the top resources found by our annotation were not amongst the top documents returned by AltaVista. For example, for the query “*human rights*,” our annotation finds the “United Nations Human Rights Website” as a *breakthrough*, while it is the twenty-second document returned by AltaVista. Thus, in its simplest form, our annotation provides a rearrangement of the results returned by AltaVista. Such rearrangements are important, since user studies have shown that the users rarely go beyond the top 20 documents returned by a web search engine [22].

CHOICE OF THE WEIGHTS

In the end, it is really the annotation of each cluster in terms of the above six nuggets that is presented to the end user.

Keywords	query: virus , Cluster 1, size = 146	query: "human rights," Cluster 1, size = 157
Summary	viruse,anti,software,information,computer,update	human,international,unit,information,nation,report
Review	Anti-Virus Tools (51)	Links To Other Human Rights Sources (40)
Breakthrough	SARC Virus EncyclopediaQ - Qm (19)	Derechos Human Rights - contact info (59)
Reference	SARC Virus EncyclopediaXn - Xz (26)	United Nations Human Rights Website (22)
Citation	McAfee.com - The Place for Your PC	Derechos - Human Rights
	Zaujimave linky	HUMAN RIGHTS REPORTING: Primary Web ...

Keywords	query: dilbert , Cluster 1, size = 165	query: terrorism , Cluster 1, size = 154
Summary	adam,book,scott,comic,work,dogbert	terrorist,state,international,attack,bomb,security
Review	DILBERT ZONE - scott adams past ... (129)	US Policy on Terrorism..Part I* (21)
Breakthrough	DILBERT ZONE - dnrc sock puppets (103)	Terrorism Research Center: Counterterrorist ... (34)
Reference	July 1995: [BUBBA-L:26422] Re: Dilbert (121)	Terrorism Research Center: Terrorist Profiles (28)
Citation	Dilbert Zone	http://www.state.gov/www/global/terrorism/
	Dilbert : On the Net 700 Sites!	Terrorism - U.S. News Net Links (116)

Table 2: By treating the entire set \mathcal{Q} as one cluster, we present the corresponding six nuggets for each of the four queries: *virus*, "human rights," *dilbert*, and *terrorism*. Every document that is in \mathcal{Q} is followed by a parenthetic number that represents its rank in the documents returned by AltaVista. Every *summary*, *review*, and *breakthrough* is always followed by a parenthetic number, whereas the *references* or *citations* are followed by a parenthetic number only when applicable. Also, see: www.almaden.ibm.com/cs/people/dmodha/toric/toric.html

Hence, arguably, a natural goal of hypertext clustering is to obtain the most descriptive and discriminative nuggets possible. Clearly, if we use $\alpha_d = 1$ and $\alpha_f = \alpha_b = 0$, then we get a good discrimination amongst the resulting clusters in the feature space constituted by the words. Consequently, we obtain good *summary* and *keywords* for the resulting clusters. Similarly, if we use $\alpha_f = 1$ and $\alpha_d = \alpha_b = 0$, then we can obtain good *review* and *references* for the resulting clusters. Finally, if we use $\alpha_b = 1$ and $\alpha_d = \alpha_f = 0$, then we can obtain good *breakthrough* and *citations* for the resulting clusters. To truly and completely exploit the hypertext nature of the given document collection, we would like all the six nuggets to be of good quality *simultaneously*. This can be achieved by judiciously selecting the parameters α_d , α_f , and α_b . We now provide a formal framework for this choice.

Throughout this section, fix the number of clusters $k \geq 2$. As before, let α_d , α_f , and α_b be nonnegative numbers that sum to 1. Geometrically, these parameters lie on a planar triangular region, say, Δ_0 , that is shown in Figure 2. For brevity, we write $\alpha = (\alpha_d, \alpha_f, \alpha_b)$. Let $\Pi(\alpha) = \{\pi_j^*\}_{j=1}^k$ denote the partitioning obtained by running the toric k -means algorithm with the parameter values α_d , α_f , and α_b . From the set of all possible clusterings $\{\Pi(\alpha) : \alpha \in \Delta_0\}$, we would like to select a partitioning that yields the *best* cluster annotations. Towards this goal, we now introduce a figure-of-merit for evaluating and comparing various clusterings.

Fix a clustering $\Pi(\alpha)$. For the given clustering, the *summary*, which is a descriptive characteristics, for each of the clusters will be good if each cluster is as coherent as possible in the word feature space, that is, if the following is maxi-

mized:

$$\Gamma_d(\alpha) \equiv \Gamma_d(\Pi(\alpha)) = \sum_{j=1}^k \sum_{x \in \pi_j} D^T D_j^*,$$

where $x = (D, F, B)$. Furthermore, the *keywords*, which are a discriminative characteristics, will be good if the following is minimized:

$$\Lambda_d(\alpha) \equiv \Lambda_d(\Pi(\alpha)) = \frac{1}{k-1} \sum_{j=1}^k \sum_{x \in \pi_j} \sum_{\ell=1, \ell \neq j}^k D^T D_\ell^*,$$

where $x = (D, F, B)$. Intuitively, $\Gamma_d(\alpha)$ and $\Lambda_d(\alpha)$ capture the *average within cluster coherence* and *average between cluster coherence*, respectively, of the clustering $\Pi(\alpha)$ in the word feature space. The *summary* and the *keywords* both will be good if the following ratio is maximized:

$$\mathcal{Q}_d(\alpha) \equiv \mathcal{Q}_d(\Pi(\alpha)) = \begin{cases} \left(\frac{\Gamma_d(\alpha)}{\Lambda_d(\alpha)} \right)^{n_d/n} & \text{if } \Lambda_d(\alpha) > 0, \\ 1 & \text{if } \Lambda_d(\alpha) = 0, \end{cases} \quad (8)$$

where n_d denotes the number of document triplets in \mathcal{Q} that have a non-zero word feature vector; see, for example, Table 1. In the case that $\Lambda_d(\alpha) = 0$, the clusters are *perfectly separated* in the word feature space.

The quantities $\Gamma_f(\alpha)$, $\Lambda_f(\alpha)$, $\Gamma_b(\alpha)$, $\Lambda_b(\alpha)$, $\mathcal{Q}_f(\alpha)$, and $\mathcal{Q}_b(\alpha)$ are defined in a similar fashion. The quantity $\mathcal{Q}_f(\alpha)$ should be maximized to obtain good quality *review* and *references*, and the quantity $\mathcal{Q}_b(\alpha)$ should be maximized to obtain good quality *breakthrough* and *citations*.

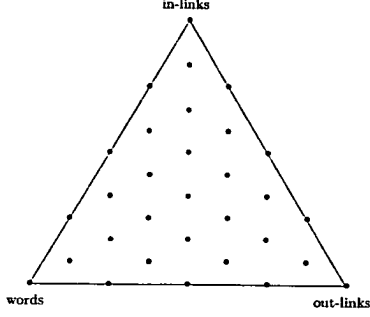


Figure 2: The triangular region Δ_0 formed by the intersection of the plane $\alpha_d + \alpha_f + \alpha_b = 1$ with the nonnegative orthant of R^3 . The left-vertex, the right-vertex, and the top-vertex of the triangle corresponds to the points $(1, 0, 0)$, $(0, 1, 0)$, and $(0, 0, 1)$, respectively.

We are now ready to present a scheme to select the optimal parameter tuple α^\dagger and the corresponding clustering $\Pi(\alpha^\dagger)$.

Step 1 Theoretically, we would like to run the toric k -means algorithm for every parameter tuple α in:

$$\Delta_0 = \{\alpha : \alpha_d + \alpha_f + \alpha_b = 1, \alpha_d, \alpha_f, \alpha_b \geq 0\}. \quad (9)$$

In practice, we replace the region Δ_0 in (9) by a finite number of points on a discrete grid that are graphically shown using the symbol \bullet in Figure 2.

Step 2 To obtain good cluster annotations in terms of all the six nuggets, we would like to simultaneously maximize Q_d , Q_f , and Q_b . Hence, we select the parameters α^\dagger as the solution of the following maximization problem:

$$\alpha^\dagger = \arg \max_{\alpha \in \Delta} [Q_d(\alpha) \times Q_f(\alpha) \times Q_b(\alpha)], \quad (10)$$

where we now define the region Δ . First, we need some notation.

$$\begin{aligned} R_d &= \{\alpha \in \Delta_0 : \Lambda_d(\alpha) = 0\} \\ R_f &= \{\alpha \in \Delta_0 : \Lambda_f(\alpha) = 0\} \\ R_b &= \{\alpha \in \Delta_0 : \Lambda_b(\alpha) = 0\} \\ \Delta_3 &= R_d \cap R_f \cap R_b \\ \Delta_2 &= ((R_d \cap R_f) \cup (R_d \cap R_b) \cup (R_f \cap R_b)) \setminus \Delta_3 \\ \Delta_1 &= (R_d \cup R_f \cup R_b) \setminus \Delta_2 \end{aligned}$$

We now define the region Δ as follows:

$$\Delta = \begin{cases} \Delta_3 & \text{if } \Delta_3 \neq \emptyset, \\ \Delta_2 & \text{elseif } \Delta_2 \neq \emptyset, \\ \Delta_1 & \text{elseif } \Delta_1 \neq \emptyset, \\ \Delta_0 & \text{otherwise.} \end{cases}$$

We now intuitively explain the reasoning behind the above definitions. The regions R_d , R_f , and R_b denote the set of

parameters for which the corresponding clusterings perfectly separate the document triplets in the word, out-link, and in-link feature spaces, respectively. The region Δ_3 denotes the set of parameters for which the corresponding clusterings perfectly separate the document triplets in *all* the three feature spaces. Clearly, if such clusterings are available, that is, if Δ_3 is not empty, then we would prefer them. Hence, we set $\Delta = \Delta_3$, if $\Delta_3 \neq \emptyset$. The region Δ_2 denotes the set of parameters for which the corresponding clusterings perfectly separate the document triplets along *two*, but not all three, feature spaces. In the case that Δ_3 is empty, we prefer clusterings in Δ_2 . Now, the region Δ_1 denotes the set of parameters for which the corresponding clusterings perfectly separate the document triplets along *one and only one* of the three feature spaces. In the case that Δ_3 and Δ_2 are both empty, we prefer the clusterings in Δ_1 . Finally, Δ_0 which is the entire triangular region in Figure 2 is the default choice when Δ_3 , Δ_2 , and Δ_1 are all empty. In practice, we have found that Δ_3 and Δ_2 are usually empty, and, hence, for most data sets, we expect the Δ to be either Δ_1 or Δ_0 .

Step 3 Let $\Pi(\alpha^\dagger)$ denote the optimal partitioning obtained by running the toric k -means algorithm with α^\dagger .

To illustrate the above scheme, we now present the Q_d , Q_f , Q_b , and $T = Q_d \times Q_f \times Q_b$ values for various parameter tuples, where \mathcal{Q} is the set of documents returned by AltaVista in response to the query *guinea* and $k = 3$.

α_d	α_f	α_b	Q_d	Q_f	Q_b	T
0.990	0.010	0.000	4.20	4.40	3.13	58.18
0.010	0.990	0.000	3.61	6.45	3.24	75.65
0.010	0.000	0.990	3.92	5.92	10.09	234.94
0.010	0.495	0.495	3.73	11.35	7.40	314.55

The first, second, and the third rows correspond to clustering primarily along words, out-links, and in-links, respectively, while the fourth row corresponds to the clustering corresponding to the optimal parameter tuple. It can be seen that the optimal clustering achieves significantly larger T value than clusterings which cluster only along one of the three features. In practice, the larger T value often translates into superior cluster annotation and a better clustering.

RESULTS: THE PROOF IS IN THE PUDDING

In Table 3, we present the parameter tuples obtained by solving the maximization problem in (10) for each of the four queries: *latex*, *abduction*, *guinea*, and *abortion*.

In Table 4, we present the optimal clusterings corresponding to the optimal parameter tuples in Table 3 for the queries *latex*, *abduction*, *guinea*, and *abortion*. It can be seen from Table 4 that (i) the set of documents corresponding to *latex* is neatly partitioned into “*latex allergies*” cluster and into “*TEX & LATEX*” cluster; (ii) the set of documents corresponding to *abduction* is neatly partitioned into “*alien abduction*”

query	k	Δ	α_d^\dagger	α_f^\dagger	α_b^\dagger
latex	2	Δ_1	0.010	0.000	0.990
abduction	2	Δ_1	0.495	0.010	0.495
guinea	3	Δ_0	0.010	0.495	0.495
abortion	3	Δ_0	0.010	0.495	0.495

Table 3: The set of documents returned by AltaVista for each of the four queries: *latex*, *abduction*, *guinea*, and *abortion* are clustered into k clusters. For each query, we determine the optimal parameter tuple $\alpha^\dagger = (\alpha_d^\dagger, \alpha_f^\dagger, \alpha_b^\dagger)$ by solving the maximization problem in (10). For queries *abduction* and *guinea*, all the three sets Δ_3 , Δ_2 , and Δ_1 turn out to be empty, and, hence, $\Delta = \Delta_0$. For queries *latex* and *abortion*, the two sets Δ_3 and Δ_2 turn out to be empty, but the set Δ_1 is not empty, and, hence, $\Delta = \Delta_1$.

cluster and into “child abduction” cluster; (iii) the set of documents corresponding to *guinea* is neatly partitioned into “Papua New Guinea,” “Guinea Bissau,” and “Guinea pigs” clusters; and, finally, (iv) the set of documents corresponding to *abortion* is neatly partitioned into two “pro-life” clusters and one “pro-choice” cluster.

FUTURE WORK

Throughout this paper, we assumed that the number of clusters k is given; however, an important future problem is to automatically determine the number of clusters in an adaptive or data-driven fashion using information-theoretic criteria such as the MDL principle.

To determine the optimal parameter tuple α^\dagger , in this paper, we run the toric k -means algorithm for every α on a certain discrete grid in the triangular region Δ_0 . We are currently investigating a computationally efficient gradient ascent procedure for computing the optimal parameter tuple α^\dagger . The basic idea is to combine the optimization problems in (10) and in (6) into a single problem that can be solved using an iterative hill-climbing heuristic.

In this paper, we have employed the new similarity measure S in the k -means algorithm; it is also possible to use it with a graph-based algorithm such as the complete link method or with hierarchical agglomerative clustering algorithms [10].

LITERATURE REVIEW

Document clustering using only textual features such as words or phrases has been extensively studied; for a detailed review of various k -means type algorithms, graph theoretical algorithms, and hierarchical agglomerative clustering algorithms, see Rasmussen [19] and Willet [26].

By treating the references made by one scientific paper (or a patent or a law case) to another as a logical hyperlink, one can interpret scientific literature (or patents or law cases) as a hypertext document collection. Citation analysis was de-

veloped as a tool to identify core sets or clusters of articles, authors, or journals of particular fields of study by using the logical hyperlinks between scientific papers, see White and McCain [25] and Small [23]. Larson [15] has proposed using citation analysis with multidimensional scaling to identify clusters in the web. Recently, Kleinberg [13] has extended citation analysis to web searching. In response to a broad-topic query, his algorithm HITS produces two distinct but inter-related types of pages: *authorities* (highly-cited pages) and *hubs* (pages that cite many authorities). HITS only uses the link topology; CLEVER refines HITS to include query word matches within anchor text [5]. For a highly accessible treatment of the use of citation analysis in web searching, see [4]. The fundamental motivation behind this paper was to seek a synthesis of text-based clustering algorithms in [19, 26, 9] and links-based eigen-analysis in [13, 5, 4]. Conceptually, our *references* and *breakthrough* are analogous to *authorities*, and our *citations* and *review* are analogous to *hubs*.

Hypertext has been used to improve information retrieval. Salton [20] has proposed using bibliographic information, that is, out-links or references, for improving retrieval performance. The basic idea is to extract important terms from cited documents and to add these non-local terms to the citing document. This line of investigation and its variants has been explored in Kwok [14], Croft and Turtle [8], Frei and Steiger [11], and, most recently, in Chakrabarti, Dom, and Indyk [3]. Our work differs from this body of work in the important aspect that we consider the out-links and the in-links as first-class features in their own right and do not use non-local terms from either the cited or citing documents. Furthermore, this body of work has not focussed on hypertext clustering which is the problem of interest in this paper.

Botafogo [1] has proposed a graph-based algorithm for clustering hypertext that uses link information but no textual information; he proposed the number of independent paths between nodes as a measure of similarity. Mukherjee, Foley, and Hudson [17] have proposed using content- and structure-based algorithms for interactive clustering of hypertext. In their model, the user precisely specifies her information need, for example, all nodes containing some content or all graphical substructures, and, hence, unlike ours, theirs is not an automated clustering methodology.

Weiss et al. [24] combined information about document contents and hyperlink structures to automatically cluster hypertext documents. While our work is closest in spirit to [24], the two works are distinct in the choice of the algorithms, the underlying similarity metrics, and the cluster naming or annotation scheme. In particular, [24] uses the complete link algorithm, while we develop a variant of the k -means algorithm. The complete link algorithm is quadratic-time complexity in the number of documents, while our method is linear-time complexity in the number of documents. Furthermore, their measure of similarity between two documents does not constitute a valid metric, and, hence, is not useful in a geometric

Keywords	query: latex , Cluster 1, size = 82	query: abduction , Cluster 1, size = 85
Summary	latex,glove,request,allergy,balloon,rubber	alien,ufo,story,experience,hip,generator
Review	Latex Allergy Injuries - The Law Offices (122)	Wiendog's Alien Abduction Page (192)
Breakthrough	Enlarger Latex Mattresses - 1(800)FloBeds (188)	What is an alien abduction experience? (116)
Reference	Latex Allergy Injuries - The Law Offices (122)	Alien Abduction Experience and Research (60)
Citation	www.FloBeds.com 1(800)FloBeds	ABIOGENESIS - POWER OF CREATION
	LATEX ALLERGY	Orthopaedic Rehabilitation. Abduction Pillows (141)

Keywords	query: latex , Cluster 2, size = 66	query: abduction , Cluster 2, size = 71
Summary	tex,document,package,command,math,postscript	child,children,parent,international,information,court
Review	Intro to TeX; LaTeX; BibTeX and SliTeX (78)	England & Wales - International ... Abduction (58)
Breakthrough	TeX and LaTeX (1)	A Halloween Abduction prevention page (105)
Reference	Peter's TeX/LaTeX/LaTeX2e/LaTeX3 Page (38)	Iran - International Parental Child Abduction (159)
Citation	TeX Frequently Asked Questions	Islamic Family Law - International ... Abduction (3)
	PROGRAMMING: bookmarks	Child Abduction - Divorce Support Net Links

Keywords	query: guinea , Cluster 1, size = 92	query: abortion , Cluster 1, size = 72
Summary	papua,country,png,weather,service,unit	life,pro,birth,partial,issue,request
Review	Papua New Guinea Map (91)	Abortion (OU CALL) (79)
Breakthrough	Weather ... Papua New Guinea Forecast (17)	Medical Misinformation About Abortion (103)
Reference	@datec ... Papua New Guinea (46)	Resource: Abortion-A Decision for Death (64)
Citation	@datec Internet - Papua New Guinea	National Right to Life Committee Main Page
	PAPUA NEW GUINEA ORCHID NEWS	http://www.learnusa.org/articles/

Keywords	query: guinea , Cluster 2, size = 34	query: abortion , Cluster 2, size = 45
Summary	pig,pigs,request,cavy,nance,live	women,cancer,baby,pregnancy,breast,heal
Review	Guinea Pig Links (196)	Project Rachel; Post-Abortion Healing ... (21)
Breakthrough	Todd's Guinea Pig Hutch (6)	Abortion; The Pontifical Academy for Life (135)
Reference	Greg's Guinea Pigs (40)	Ohio Abortion Statistics (102)
Citation	Todd's Guinea Pig Hutch (6)	Life Institute-Proclaiming The Gospel of Life ...
	OinkerNet & Guinea Pigs Worldwide!	Abortion References, Statistics; Study; Research ...

Keywords	query: guinea , Cluster 3, size = 20	query: abortion , Cluster 3, size = 39
Summary	bissau,travel,information,embassy,island,world	reproductive,action,error,clinic,information,caral
Review	Guinea Bissau @ Travel Notes (r). (70)	California Abortion & Reproductive (CARAL) (138)
Breakthrough	Papua New Guinea @ Travel Notes (r). (160)	California Abortion & Reproductive (CARAL) (35)
Reference	Guinea-Bissau; with National Anthem ... (23)	China: Abortion (43)
Citation	Country Information @ ... Online Travel Guide.	Reproductive Health & Rights Center: Home Page
	National Anthems of the World.	Dr. Pranikoff's Gyn Web Library - Abortion

Table 4: By running the toric k -means algorithm with the respective optimal parameter tuples in Table 3, we cluster the set of documents \mathcal{Q} returned by AltaVista in response to the queries *latex*, *abduction*, *guinea*, and *abortion* into $k = 2, 2, 3$, and 3 clusters, respectively. We show the six nuggets for each cluster. Every document that is in \mathcal{Q} is followed by a parenthetic number that represents its rank in the documents returned by AltaVista. Every *summary*, *review*, and *breakthrough* is always followed by a parenthetic number, whereas the *references* or *citations* are followed by a parenthetic number only when applicable. Also, see:

www.almaden.ibm.com/cs/people/dmodha/toric/toric.html

setting like ours. Finally, our cluster annotation scheme has no analogue in [24].

Previously, Pirolli, Pitkow, and Rao [18] have combined both the link "topology and textual similarity between items as well as usage data collected by servers and page meta-information like title and size". [18] did not treat link topology and textual similarity differently as we do, but rather represented each hypertext document as a single vector of all these features. They left the problem of automatically categorizing hypertext documents using their feature space to future work. Chen [6] has proposed generalized similarity analysis that combines hypertext linkage, content similarity, and browsing patterns or usage. Chen and Czerwinski [7] have exploited generalized similarity analysis along with latent semantic indexing and pathfinder network scaling to develop an integrated framework for spatial organization of information and for browsing and searching. Their results are complementary to ours.

REFERENCES

1. BOTAFAGO, R. A. Cluster analysis for hypertext systems. In *ACM SIGIR* (1993).
2. BRADLEY, P., AND FAYYAD, U. Refining initial points for k-means clustering. In *ICML* (1998), pp. 91–99.
3. CHAKRABARTI, S., DOM, B. E., AND INDYK, P. Enhanced hypertext categorization using hyperlinks. In *ACM SIGMOD* (1998).
4. CHAKRABARTI, S., DOM, B. E., KUMAR, S. R., RAGHAVAN, P., RAJAGOPALAN, S., TOMKINS, A., KLEINBERG, J. M., AND GIBSON, D. Hypersearching the web. *Scientific American* (June 1999).
5. CHAKRABARTI, S., DOM, B. E., RAGHAVAN, P., RAJAGOPALAN, S., GIBSON, D., AND KLEINBERG, J. Automatic resource compilation by analyzing hyperlink structure and associated text. In *WWW7* (1998).
6. CHEN, C. Structuring and visualizing the www by generalized similarity analysis. In *ACM Hypertext* (1997).
7. CHEN, C., AND CZERWINSKI, M. From latent semantics to spatial hypertext—An integrated approach. In *ACM Hypertext* (1998).
8. CROFT, W. B., AND TURTLE, H. R. A retrieval model for incorporating hypertext links. In *ACM Hypertext* (1989).
9. DHILLON, I. S., AND MODHA, D. S. Concept decompositions for large sparse text data using clustering. Tech. Rep. RJ 10147 (95022), IBM Almaden Research Center, 1999.
10. FRAKES, W. B., AND BAEZA-YATES, R. *Information Retrieval: Data Structures and Algorithms*. Prentice Hall, Englewood Cliffs, New Jersey, 1992.
11. FREI, H. P., AND STEIGER, D. Making use of hypertext links when retrieving information. In *ACM European Conference on Hypertext* (1992).
12. HARTIGAN, J. A. *Clustering Algorithms*. Wiley, 1975.
13. KLEINBERG, J. Authoritative sources in a hyperlinked environment. In *ACM-SIAM SODA* (1998).
14. KWOK, K. L. A probabilistic theory of indexing and similarity measure based on cited and citing documents. *J. Amer. Soc. Inform. Sci.* (1985), 342–351.
15. LARSON, R. Bibliometric of the world wide web: An exploratory analysis of the intellectual structure of cyberspace. In *Annual Meeting Amer. Soc. Info. Sci.* (1996).
16. LAWRENCE, S., AND GILES, C. L. Searching the World Wide Web. *Science* 280, 5360 (1998), 98.
17. MUKHERJEA, S., FOLEY, J. D., AND HUDSON, S. E. Interactive clustering for navigating in hypermedia systems. In *ACM Hypertext* (1994).
18. PIROLI, P., PITKOW, J., AND RAO, R. Silk from sow's ear: Extracting usable structures from the web. In *ACM SIGCHI Human Factors Comput.* (1996).
19. RASMUSSEN, E. Clustering algorithms. In *Information Retrieval: Data Structures and Algorithms* (1992), W. B. Frakes and R. Baeza-Yates, Eds., Prentice Hall, Englewood Cliffs, New Jersey, pp. 419–442.
20. SALTON, G. Associative document retrieval techniques using bibliographic information. *J. ACM* (1963), 440–457.
21. SALTON, G., AND MCGILL, M. J. *Introduction to Modern Retrieval*. McGraw-Hill Book Company, 1983.
22. SILVERSTEIN, C., HENZINGER, M., MARAIS, J., AND MORICZ, M. Analysis of a very large AltaVista query log. Tech. Rep. 1998-014, Compaq Systems Research Center, Palo Alto, CA, October 1998.
23. SMALL, H. Co-citation in the scientific literature: A new measure of the relationship between two documents. *J. Amer. Soc. Inform. Sci.* (1973), 265–269.
24. WEISS, R., VELEZ, B., SHELDON, M. A., NAMPREMPRE, C., SZILAGYI, P., DUDA, A., AND GIFFORD, D. K. Hypursuit: A hierarchical network search engine that exploits content-link hypertext clustering. In *ACM Hypertext* (1996).
25. WHITE, H. D., AND MCCAIN, K. W. Bibliometrics. *Annual Review of Information Science and Technology* 24 (1989), 119–186.
26. WILLET, P. Recent trends in hierarchic document clustering: a critical review. *Inform. Proc. & Management* (1988), 577–597.